

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATATYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE	CHARACTER_SET_NAME	COLLATION NAME	DTD_IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL_DATA_ACCESS	SQL_PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION
build_email_list	def	package	build_email_list	PROCEDURE									SQL	BEGIN DECLARE v_finished INTEGER DEFAULT 0; DECLARE v_email varchar(100) DEFAULT ''; -- declare cursor for employee email DECLARE email_cursor CURSOR FOR SELECT email FROM register; -- declare NOT FOUND handler DECLARE CONTINUE HAND			SQL	NO	CONTAINS SQL		DEFINER	2016-10-27 21:39:37	2016-10-27 21:39:37			root@localhost	utf8	utf8_general_ci	latin1_swedish_ci

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATATYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE	CHARACTER_SET_NAME	COLLATION NAME	DTD_IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL_DATA_ACCESS	SQL_PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION											
													LER FOR NOT FOUND SET v_finished = 1; OPEN email_cursor; get_email: LOOP FETCH email_cursor INTO v_email; IF v_finished = 1 THEN LEAVE get_email; END IF; -- build email list SET email_list = CONCAT(v_email, ";", email_list); END LOOP get_email;																											

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATATYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE	CHARACTER_SET_NAME	COLLATION NAME	DTD_IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL_DATA_ACCESS	SQL_PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL_MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION
														CLOSE email cursor ;															
build_email_list1	def_	package	build_email_list1	PROCEDURE									SQL	BEGIN			SQL	NO	CONTAINS SQL		DEFINER	2016-10-27 22:52:33	2016-10-27 22:52:33			root@localhost	utf8	utf8_general_ci	latin1_swedish_ci
														DECLARE v_finished INTEGER DEFAULT 0;															
														DECLARE v_email varchar(100) DEFAULT '';															
														DECLARE email_cursor CURSOR FOR															
														SELECT email FROM register;															
														-- declare NOT FOUND handler															
														DECLARE CONTINUE HAND															

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATATYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE	CHARACTER_SET_NAME	COLLATION NAME	DTD_IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL_DATA_ACCESS	SQL_PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION											
													LER FOR NOT FOUND SET v_finished = 1; OPEN email_cursor; get_email: LOOP FETCH email_cursor INTO v_email; IF v_finished = 1 THEN LEAVE get_email; END IF; -- build email list SET email_list = CONCAT(v_email, ";", email_list); END LOOP get_email;																											

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATATYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE	CHARACTER_SET_NAME	COLLATION NAME	DTD_IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL_DATA_ACCESS	SQL_PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION
														CLOSE email cursor; END															
GetAllProducts	def	package	GetAllProducts	PROCEDURE									SQL	BEGIN SELECT * FROM register; END			SQL	NO	CONTAINS SQL		DEFINER	2016-10-27 20:07:08	2016-10-27 20:07:08			root@localhost	utf8	utf8_general_ci	latin1_swedish_ci
laundry_bill	def	package	laundry_bill	PROCEDURE									SQL	BEGIN DECLARE bill INTEGER DEFAULT 0; declare pay integer default 0; -- declare cursor for employee email DECLARE cursor_1 CURSOR FOR SELECT type FROM laundry; -- declare			SQL	NO	CONTAINS SQL		DEFINER	2016-10-27 21:55:20	2016-10-27 21:55:20			root@localhost	utf8	utf8_general_ci	latin1_swedish_ci

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATATYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE	CHARACTER_SET_NAME	COLLATION NAME	DTD_IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL_DATA_ACCESS	SQL_PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION											
														e NOT FOUND handler DECLARE CONTINUE HANDLER FOR NOT FOUND SET bill = 1; OPEN cursor_1; get_email: LOOP FETCH cursor_1 INTO bill; IF bill = 1 THEN LEAVE get_email; END IF; -- build email list SET pay = type*5; END LOOP get_email;																										

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATATYPE	CHARACTER MAXIMUM LENGTH	CHARACTER OCTET LENGTH	NUMERIC PRECISION	NUMERIC SCALE	CHARACTER SET NAME	COLLATION NAME	DTD IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL DATA ACCESS	SQL PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION
														CLOSE cursor_1;															
laundry_bill_1	def	package	laundry_bill_1	PROCEDURE									SQL	BEGIN			SQL	NO	CONTAINS SQL		DEFINER	2016-10-27 22:03:46	2016-10-27 22:03:46			root@localhost	utf8	utf8_general_ci	latin1_swedish_ci
														DECLARE bill INTEGER DEFAULT 0;															
														declare pay integer default 0;															
														-- declare cursor for employee email															
														DECLARE cursor_1 CURSOR FOR															
														SELECT type FROM laundry;															
														-- declare NOT FOUND handler															
														DECLARE															

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATATYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE	CHARACTER_SET_NAME	COLLATION NAME	DTD_IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL_DATA_ACCESS	SQL_PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION										
														CONTINUE HANDLER FOR NOT FOUND SET bill = 1; OPEN cursor _1; get_email: LOOP FETCH cursor _1 INTO bill; IF bill = 1 THEN LEAVE get_email; END IF; -- build email list SET pay = type*5 ; END LOOP get_email; CLOSE cursor _1; END																									
proce	def	packa	proce	PROCE									SQL	BEGIN			SQL	NO	CONT		DEFIN	2016-	2016-			root@	utf8	utf8_g	latin1										

SPECIFIC NAME	ROUTINE CATALOG	ROUTINE SCHEMA	ROUTINE NAME	ROUTINE TYPE	DATA TYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE	CHARACTER_SET_NAME	COLLATION NAME	DTD_IDENTIFIER	ROUTINE BODY	ROUTINE DEFINITION	EXTERNAL NAME	EXTERNAL LANGUAGE	PARAMETER STYLE	IS DEFINISTIC	SQL_DATA_ACCESS	SQL_PATH	SECURITY TYPE	CREATED	LAST ALTERED	SQL MODE	ROUTINE COMMENT	DEFINER	CHARACTER SET CLIENT	COLLATION CONNECTION	DATABASE COLLATION
		ge		DURE										SELECT name FROM register WHERE r_id = userId ; END					AINS SQL		ER	10-27 22:38 :11	10-27 22:38 :11			localhost		enera_l_ci	swedish_ci
room number	def	package	room number	PROCEDURE									SQL BEGIN SELECT room_no FROM room WHERE status = 'A'; END			SQL	NO	CONTAINS SQL		DEFINER	2016-10-29 16:48 :43	2016-10-29 16:48 :43			root@localhost	utf8	utf8_enera_l_ci	latin1_swedish_ci	
usp_todalEmployeeByDepartment	def	package	usp_todalEmployeeByDepartment	PROCEDURE									SQL begin select count (*) as total from register where r_id = id; end			SQL	NO	CONTAINS SQL		DEFINER	2016-10-27 21:28 :30	2016-10-27 21:28 :30			root@localhost	utf8	utf8_enera_l_ci	latin1_swedish_ci	